DETAILED ACTION

Response to Arguments

Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1 – 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Berman (U.S. Patent 6,502,194) in view of Zainoulline (U.S. Patent Application Publication 2001/0030660) in further view of Hsu (U.S. Patent Application Publication 2003/0069854).

Regarding **Claim 1**,

Berman discloses:

An apparatus for smoothly playing a pre-determined sequence of streamed content segments (Fig. 1 element 100), comprising:

a processor (Fig. 1 element 118),

a first memory (116) that stores at least one control program useable by the processor to control play of the predetermined sequence of content segments (col. 6 lines 40 – 50),

wherein the apparatus is configured to:

in response to initiation of play of a content segment, initiate downloading to the pre-buffer cache of a beginning portion of a number of content segments which are, in the predetermined sequence, subsequent to the content segement (i.e. as the first song (Song 1) is being played, the playback unit continues to operate and, in background operations, continues to download the Song 1 data into the first buffer, and also downloads data for the other selected songs into the other buffers in an alternating fashion. Each song will be placed into a different sequential buffer; col. 12 lines 10 – 16; the buffers in the same memory and thus common to one another via the same memory;);

in response to skipping to a target content segment of the predetermined sequence of content segments whose beginning portion has been downloaded to the pre-buffer cache, initiate play of the downloaded beginning portion of the target content segment(i.e. this ensures that some portion of each selected song will be downloaded and available as soon as possible, thereby permitting the user to skip to one of the other selected songs after playback has begun; col. 12 lines 16 – 19);

while playing the downloaded beginning portion of the target content segment, initiated downloading of the rest of the target content segment (i.e. songs are downloaded as they are played and designated for buffers, col. 11 lines 65-67 and col. 12 lines 1 - 20).

Berman does not explicitly disclose the at least one control program including computer-readable instructions specifying a number of beginning portions of content segments to cache in advance and size of a pre-buffer cache. However, Berman recognizes that the size of the buffer and number of songs are relatively flexible. Berman clearly talks about typical sizes that can be used for buffer size and that the memory may be segregated into a "number" of sequential buffers in the Memory Buffering Control section. Berman further discusses that the functionality of the device remains the same regardless of the memory space addressed. It would have been obvious to provide the allocation and setup of this buffer in this system using software as claimed. The system disclosed by Berman is a processor/memory type system which is notoriously well known to operate and function using software. While not expressly indicated, it is likely that Berman performs these functions using software. Even if not, it would have been obvious for one of ordinary skill in the art to try given the notoriously well known components disclosed and their known interrelationships, (i.e. processor/software/memory) and the known benefits of a software system (ease of use, quick changes in operation, flexibility).

Berman does not explicitly disclose a second memory which is available to the at least one control program, or the wherein the pre-buffer cache is in an area in said second memory.

Zainoulline discloses a preview device having a CPU, RAM memory which loads the player programs, and staging memory which stores the preview clips (page 3 paragraph 0031).

Modifying Zainoulline to include this staging memory separate to the RAM storing the control programs in order to store the music data in Berman instead of Zainoulline's only memory reads upon the limitation of a second memory which is available to said at least one program for operations and wherein the pre-buffer cache is in an area in said second memory.

One of ordinary skill in the art at the time of the invention would have been motivated to use Zainoulline's preview device with Berman's Memory Buffering Control playback method in order to create a more pleasing online shopping experience. Rather than a user having to wait for each individual song to buffer as they skip between preview clips, the combination would allow a user to smoothly switch between media clips thereby saving the user time and avoiding annoying pauses between playback (Zainoulline paragraph 26).

The combination also fails to explicitly disclose initiating play of the downloaded beginning portion of the target content segment if less than a preallocated quantity of content segments were previously streamed during a subscription period, in association with a subscriber.

However, allowing a limited number of plays in a streaming music environment is notoriously well known. Hsu teaches of a device that delivers music to a playback device over a network including an associated expiration such as allowing a user to play the audio files only a certain number of times. The user can also choose to refresh the expiration or obtain new playback rights; para 0004 as well as purchase subscription programs para 0013.

Applying the teachings of Hsu to the combination teaches:

initiating play of the downloaded beginning portion of the target content segment (i.e. playback on the above combination) if less than a preallocated quantity of content segments were previously streamed during a subscription period (i.e. Modifying the combination to only allow playback of audio files a certain number of times as taught by Hsu), in association with a subscriber (i.e. user w/ a subscription).

It would have been obvious to one of ordinary skill in the art to apply the rights management system of Hsu to the combination of Berman in view of Zainoulline. The combination is namely concerned with rights management and allowing previews. Adding another layer of rights management as taught by Hsu would have been nothing more than use of a known technique to improve similar devices in the same way.

Regarding **Claims 6, 7, 16 and 17**, in addition to the elements stated above regarding claim 1, the combination further discloses:

All elements are met by the rejection of claim 1 above, except:

The combination fails to explicitly disclose wherein the number of beginning portions of content segments to pre-cache in advance and size of the pre-buffer are specified by a function call.

However, function calls within programs (such as the computer program made obvious by the combination) are notoriously well known in the art. The Microsoft Computer Dictionary defines a function call broadly as "A program's request for the services of a particular function. A function call is coded as the name of the function along with any parameters needed for the function to perform its task." Applying this to the combination, which clearly shows a flexible buffer system, would allow the program to set the number of buffers via one of these calls for the purpose of allowing the function to perform its task. Applying this known technique (i.e. a function call) to a known device (i.e. software setup for a flexible buffer system) would have been obvious to one of ordinary skill in the art. Modifying the above would have provided predictable results (i.e. a software program call to set up the number of buffers) since the relatively flexible buffer system disclosed by the combination doesn't explicitly set forth the means of implementation.

The combination further discloses;

(c) if playback is skipped from a playing content segment to a target content segment, checking whether the beginning portion of the target content segment is in the pre-buffer cache; and

(d) if the beginning portion of the target content segment is in the pre-buffer cache, playing the beginning portion of the target content segment from the pre-buffer cache (i.e. Berman further discloses checking to see if the track is in the buffer and if so beginning to stream track data from memory; Fig. 5 elements 506 and 512).

As stated above regarding claim 1, Berman discloses data in a given buffer is overwritten as it is processed and played. Thus, after the last segment of memory in a buffer for a song has been filled with a song data packet and that buffer is processed for listening, the next song data packet will be written to the first segment in that buffer (col. 12 lines 25 – 30). Therefore, as the system starts downloading the rest of the said target content segment, it is inherent that the data that has been in the buffer prior to the target segment is overwritten (i.e. deleted) as the newer data is being processed and played. This reads upon the limitation of (f) deleing beginning portions of any content segment prior to the target content segment in the predetermined sequence from the pre-buffer cache. Element (e) is met above regarding claim 1.

Regarding **Claims 2, 12 and 22**, in addition to the elements above regarding claim 1, the combination further discloses

wherein the beginning portion of the target content segment is approximately the data of the first ten seconds of the target content segment (in Berman in the preferred embodiment each data packet contains approximately ten seconds of compressed digital audio information; col. 11 lines 50 –52).

Regarding **Claim 3, 13, and 23**, in addition to the elements above regarding claim 1, the combination further discloses:

Berman discloses three buffers in a playback memory in Figure 11. The playback unit memory may be segregated into a number of sequential buffers, with each buffer preferably containing one song (col. 11 lines 30 – 32) and the number of buffers is determined by the 2MB buffer size and the amount of memory that the playback unit microprocessor can access, so the number of buffers available will be variable (col. 11 lines 34 –38). Since microprocessor accessible memories of, for example, 256 MB, are well known at the time of the invention, Berman's disclosure comprehends any number of buffers up to at least 128.

Regarding **Claims 4, 14, and 24**, in addition to the elements stated above regarding claim 1, the combination further discloses:

Berman further discloses that the buffers correspond to the following musical selections (col. 11 lines 63 – 65) and that the buffers are sequential buffers (col. 11 line 31). Berman discloses that the buffers correspond to the following musical selections as well as hold the data of the following songs to be played in sequential order. Therefore, it is taught that the number of beginning portions of content segments to cache in advance is all content segments in the predetermined sequence of content segments that are subsequent to the playing content segment.

Regarding **Claims 5, 15, and 25**, in addition to the elements stated above regarding claim 1, the combination further discloses:

wherein the pre-buffer cache follows a first-in first-out algorithm and allows writing while reading (i.e. Berman further discloses The loop buffering operation progresses from left to right in Fig 12. Loop buffering is used to limit the size needed for each buffer. In particular, a buffer is not expected to have sufficient capacity to contain the entire data needed for one song. Rather data in a given buffer is overwritten as it is processed and played. Thus, after the last segment of memory in a buffer for a song has been filled with a song data packet and that buffer is processed for listening, the next song data packet will be written to the first segment in that buffer; col. 12 lines 22 – 30).

Regarding **Claims 8 and 18**, in addition to the elements stated above regarding claims 7 and 17, the combination further discloses:

If the beginning portion for the target content segment is in the pre-buffer cache, downloading, consecutively, a beginning portion of each of a number of content segments which are, in the predetermine sequence, subsequent to the target content segment, wherein if beginning portions of the one or more content segments subsequent to the target content segment are already in the pre-buffer cache, skipping the downloading of the beginning portions of the one or more content segments already having beginning portions in the pre-buffer cache and downloading the beginning portions of the subsequent content segments such that beginning portions of each of the number of content segments to cache in advance are downloaded to the pre-buffer cache.

Berman discloses that portions of each selected song will be downloaded as the first one begins to play (col. 11 lines 56 and 57), the number of buffers may be variable (col. 11 lines 37 and 38), this ensures that some portion of each selected song will be downloaded and available as soon as possible, thereby permitting the user to skip to one of the other selected songs after playback has begun (col.12 lines 16 – 19), and as the first song (Song 1) is being played, the playback unit continues to operate and, in background operations, continues to download the Song 1 data into the first buffer, and also downloads data for the other selected songs into the other buffers in an alternating fashion. Each song will be placed into a different sequential buffer. (col.12 lines 10 – 16). Thus it is taught that as soon as the user skips ahead to another song, the subsequent songs will be downloaded into the buffer sequentially in order to fill the number of buffers as the system downloads the portions of the other songs not playing as shown above, as well as downloading the portions for the next songs in advance.

Regarding **Claims 9 and 19**, in addition to the elements stated above regarding claims 8 and 18, the combination further discloses:

if no skip command is received while the target content segment is playing, as the playing of the target content segment ends, playing song immediately subsequent to the target content segment.

Berman further discloses that if a user wants to hear Song1, Song2, and Song 3, the playback unit downloads a number of packets for Song1 into the first available buffer, Once a sizeable amount of compressed audio information is stored for that song, the playback unit begins to process the information and play the song (col.11 lines 66 and 67, col. 12 lines 1-4). It is inherent that if the user selects these three songs, starts playing Song1, and doesn't skip ahead that Song 2 will follow after Song1 has completed playing based on the functionality of the buffer.

The remaining limitations are met by the rejection of claim 7 above.

Regarding **Claims 10 and 20**, in addition to the elements stated above regarding claims 7 and 17, the combination further discloses:

wherein if the beginning portion of the target content segment is not in the pre-buffer cache, the method further comprises:

(k) sending request to stop transmitting the playing content segment and to start transmitting the target content segment and at least substantially simultaneously and (n) begin playing the target content segment as a sufficient portion of the target content segment has been downloaded.

Berman discloses that As the first song (Song 1 is being played, the playback unit continues to operate and, in background operation, continues to download the Song 1 data into the first buffer, and also downloads data for the other selected songs in to the other buffers into an alternating fashion (col. 12 lines 10 – 14) and if a user wants to hear Song1, Song2, and Song 3, the playback unit downloads a number of packets for Song1 into the first available buffer, Once a sizeable amount of compressed audio information is stored for that song, the playback unit begins to process the information and play the song (col.11 lines 66 and 67, col. 12 lines 1-4). Therefore, if a user starts playing Song1 and instantly skips to Song2 there will be no information stored in the buffer for Song2 therefore it is inherent that the system will stop playing Song 1 and automatically download the information for Song2.

Elements (l) (deleting pre cached like (f) in another playback) and (m) (downloading like (e) in another playback) are met by the rejection of claim 7 as stated above.

Regarding **Claims 11 and 21**, in addition to the elements stated above regarding claims 10 and 20, further use of the system allows for additional skips, pauses, plays etc and thus, element (p) is clearly comprehended above regarding claim 9 element (j), element (q) is clearly comprehended above regarding claim 9 element (i), element (r) is clearly comprehended above regarding claim 7 element (e), element (s) is clearly comprehended above regarding claim 8 element (h), and element (t) is clearly comprehended above regarding claim 7 element (g).

Claim 26 is met by the rejections of claims 1, 6 and 7 as stated above.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ANDREW C. FLANDERS whose telephone number is (571)272-7516. The examiner can normally be reached on M-F 8:30 - 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Curtis Kuntz can be reached on (571) 272-7499. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Andrew C Flanders/
Primary Examiner, Art Unit 2614